Nested Canalyzing, Unate Cascade, and Polynomial Functions [1]

Abdul Salam Jarrah [*,a] Blessilda Raposa [b]
Reinhard Laubenbacher [a]

[a] *Virginia Bioinformatics Institute (0477), Virginia Tech, Blacksburg, VA 24061, USA*

[b] *Mathematics Department, De La Salle University, 2401 Taft Avenue, Manila, Philippines*

Abstract

This paper focuses on the study of certain classes of Boolean functions that have appeared in several different contexts. Nested canalyzing functions have been studied recently in the context of Boolean network models of gene regulatory networks. In the same context, polynomial functions over finite fields have been used to develop network inference methods for gene regulatory networks. Finally, unate cascade functions have been studied in the design of logic circuits and binary decision diagrams. This paper shows that the class of nested canalyzing functions is equal to that of unate cascade functions. Furthermore, it provides a description of nested canalyzing functions as a certain type of Boolean polynomial function. Using the polynomial framework one can show that the class of nested canalyzing functions, or, equivalently, the class of unate cascade functions, forms an algebraic variety which makes their analysis amenable to the use of techniques from algebraic geometry and computational algebra. As a corollary of the functional equivalence derived here, a formula in the literature for the number of unate cascade functions provides such a formula for the number of nested canalyzing functions.

Key words: nested canalyzing function, unate cascade function, parametrization, polynomial function, Boolean function, algebraic variety

* The corresponding author.
Email addresses: `ajarrah@vbi.vt.edu` (Abdul Salam Jarrah), `raposab@dlsu.edu.ph` (Blessilda Raposa), `reinhard@vbi.vt.edu` (Reinhard Laubenbacher).

[1] The authors thank the referee for the constructive and insightful comments which have improved this paper significantly.

1 Introduction

Canalyzing functions were introduced by Kauffman [10] as appropriate rules in Boolean network models of gene regulatory networks. The definition is reminiscent of the concept of "canalisation" introduced by the geneticist Waddington [22] to represent the ability of a genotype to produce the same phenotype regardless of environmental variability. Canalyzing functions are known to have other important applications in physics, engineering and biology. They have been used to study the convergence behavior of a class of nonlinear digital filters, called stack filters, which have applications in image and video processing [5; 23; 24]. Canalyzing functions also play an important role in the study of random Boolean networks [10; 13; 19; 20], have been used extensively as models for dynamical systems as varied as gene regulatory networks [10], evolution, [20] and chaos [13]. One important characteristic of canalyzing functions is that they exhibit a stabilizing effect on the dynamics of a system. For example, in [5], it is shown that stack filters which are defined by canalyzing functions converge to a fixed point called a root signal after a finite number of passes. Moreira and Amaral [15], showed that the dynamics of a Boolean network which operates according to canalyzing rules is robust with regard to small perturbations.

A special type of canalyzing function, so-called *nested canalyzing functions* (NCFs) were introduced recently in [8], and it was shown in [9] that Boolean networks made from such functions show stable dynamic behavior and might be a good class of functions to express regulatory relationships in biochemical networks. Little is known about this class of functions, however. For instance, there is no known formula for the number of nested canalyzing functions in a given number of variables.

Another field in which special families of Boolean functions have been studied extensively is the theory of computing, in particular the design of efficient logical switching circuits. Since the 1970s, several families of Boolean functions have been investigated for use in circuit design. For instance, the family of *fanout-free* functions has been studied extensively, as well as the family of cascade functions. A subclass of these are the *unate cascade functions* see, e.g., [14; 16], which we focus on here. It turns out that this class of functions has some very useful properties. For instance, it was shown recently [3] that the class of unate cascade functions is precisely the class of Boolean functions that have good properties as binary decision diagrams. In particular, the unate cascade functions (on n variables) are precisely those functions whose binary decision diagrams have the smallest average path length $(2 - \frac{1}{2^{n-1}})$ among all Boolean functions of n variables.

The notion of average path length is one cost measure for binary decision trees, which measures the average number of steps to evaluate the function on which the tree is based. One way of assessing the relative efficacy of classes of Boolean function for logic circuit or binary decision tree design is to look at the number of different circuits or trees that can be realized with a particular class. That is, one would like to count the number of functions in a given class. This has led to a formula for the number of unate cascade functions [2]. One of the results in this paper shows that the classes of unate cascade functions and nested canalyzing functions are

identical (as classes of functions rather than as classes of logical expressions). As a result of the equivalence we will establish, this formula then also counts the number of nested canalyzing functions.

A third framework for studying Boolean functions, in the context of models for biochemical networks, was introduced in [11]. There, a new method to reverse engineer gene regulatory networks from experimental data was proposed. The proposed modeling framework is that of time-discrete deterministic dynamical systems with a finite set of states for each of the variables. The number of states is chosen so as to support the structure of a finite field. One consequence is that each of the state transition functions can be represented by a polynomial function with coefficients in the finite field, thereby making available powerful computational tools from polynomial algebra. This class of dynamical systems in particular includes Boolean networks, when network nodes take on two states. It is straightforward to translate Boolean functions into polynomial form, with multiplication corresponding to AND, addition to XOR, and addition of the constant 1 to negation. In this paper we provide a characterization of those polynomial functions over the field with two elements that correspond to nested canalyzing (and, therefore, unate cascade) functions. Using a parameterized polynomial representation, one can characterize the parameter set in terms of a well-understood mathematical object, a common method in mathematics. This is done using the concepts and language from algebraic geometry. To be precise, we describe the parameter set as an algebraic variety, that is a set of points in an affine space that represents the set of solutions of a system of polynomial equations. This algebraic variety turns out to have special structure that can be used to study the class of nested canalyzing functions as a rich mathematical object.

2 Boolean Nested Canalyzing and unate cascade Functions are equivalent

2.1 Boolean Nested Canalyzing Functions

Boolean nested canalyzing functions were introduced recently in [8], and it was shown in [9] that Boolean networks made from such functions show stable dynamic behavior. In this section we show that the set of Boolean nested canalyzing functions is equivalent to the set of unate cascade functions that has been studied before in the engineering and computer science literature. In particular, this equivalence provides a formula for the number of nested canalyzing functions in a given number of variables.

We begin by defining the canalyzing property.

Definition 2.1 *A Boolean function $f(x_1, \ldots, x_n)$ is* canalyzing *if there exists an index i and a Boolean value a for x_i such that $f(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n) = b$ is constant. That is, the variable x_i, when given the* canalyzing value *a, determines the value of the function f, regardless of the other inputs. The output value b is called the* canalyzed value.

Throughout this paper, we use the Boolean functions $\mathrm{AND}(x, y) = x \wedge y$, $\mathrm{OR}(x, y) = x \vee y$ and $\mathrm{NOT}(x) = \overline{x}$.

Example 2.2 *The function* $\mathrm{AND}(x, y) = x \wedge y$ *is a canalyzing function in the variable* x *with canalyzing value 0 and canalyzed value 0. The function* $\mathrm{XOR}(x, y) := (x \vee y) \wedge \overline{(x \wedge y)}$ *is not canalyzing in either variable.*

Nested canalyzing functions are a natural specialization of canalyzing functions. They arise from the question of what happens when the function does not get the canalyzing value as input but instead has to rely on its other inputs. Throughout this paper, when we refer to a function of n variables, we mean that f depends on all n variables. That is, for $1 \leq i \leq n$, there exists $(a_1, \ldots, a_n) \in \mathbb{F}_2^n$ such that $f(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, \overline{a_i}, a_{i+1}, \ldots, a_n)$.

Definition 2.3 *Let* f *be a Boolean function in* n *variables.*

- *Let* σ *be a permutation on* $\{1, \ldots, n\}$. *The function* f *is* nested canalyzing function*(NCF) in the variable order* $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ *with canalyzing input values* $a_1, \ldots, a_n$ *and canalyzed output values* $b_1, \ldots, b_n$, *respectively, if it can be represented in the form*

$$f(x_1, x_2, \ldots, x_n) = \begin{cases} b_1 & \text{if } x_{\sigma(1)} = a_1, \\ b_2 & \text{if } x_{\sigma(1)} \neq a_1 \text{ and } x_{\sigma(2)} = a_2, \\ b_3 & \text{if } x_{\sigma(1)} \neq a_1 \text{ and } x_{\sigma(2)} \neq a_2 \text{ and } x_{\sigma(3)} = a_3, \\ \vdots & \qquad \vdots \\ b_n & \text{if } x_{\sigma(1)} \neq a_1 \text{ and } \cdots \text{ and } x_{\sigma(n-1)} \neq a_{n-1} \text{ and } x_{\sigma(n)} = a_n, \\ \overline{b_n} & \text{if } x_{\sigma(1)} \neq a_1 \text{ and } \cdots \text{ and } x_{\sigma(n)} \neq a_n. \end{cases} \tag{2.1}$$

- *The function* f *is nested canalyzing if* f *is nested canalyzing in the variable order* $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ *for some permutation* σ.

Example 2.4 *The function* $f(x, y, z) = x \wedge \overline{y} \wedge z$ *is nested canalyzing in the variable order* x, y, z *with canalyzing values 0,1,0 and canalyzed values 0,0,0, respectively. However, the function* $f(x, y, z, w) = x \wedge y \wedge \mathrm{XOR}(z, w)$ *is not nested canalyzing because if* $x = 1$ *and* $y = 1$, *then the value of the function is not constant for any input values for either* z *or* w.

The following lemma follows directly from the definition above.

Lemma 2.5 *A Boolean function* f *on* n *variables is nested canalyzing in the variable order* $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ *with canalyzing input values* $a_1, \ldots, a_n$ *and canalyzed output values* $b_1, \ldots, b_n$, *respectively if and only if* f *depends on all* n *variables and, for all* $1 \leq i \leq n$,

$$f(c_1, \ldots, c_n) = b_i,$$

where $(c_1, \ldots, c_n) \in \mathbb{F}_2^n$ *such that* $c_{\sigma(i)} = a_i$ *and, for* $1 \leq j < i$, $c_{\sigma(j)} = \overline{a_j}$.

The next lemma gives the functional form of a nested canalyzing function. To simplify notation, we will use the following notational convention. Let $a = 0, 1$. Then $x + a$ will denote x

if $a = 0$ and $\overline{x}$ if $a = 1$.

Lemma 2.6 *Let*

$$g(x_1, \ldots, x_n) = (x_{\sigma(1)}+a_1+b_1)\Diamond_1((x_{\sigma(2)}+a_2+b_2)\Diamond_2(\cdots((x_{\sigma(n-1)}+a_{n-1}+b_{n-1})\Diamond_{n-1}(x_{\sigma(n)}+a_n+b_n))\cdots), \tag{2.2}$$

where

$$\Diamond_i = \begin{cases} \vee, & \text{if } b_i = 1; \\ \wedge, & \text{if } b_i = 0, \end{cases} \tag{2.3}$$

and $a_i, b_i \in \{0,1\}$ for all $1 \le i \le n$. Then g is nested canalyzing in the variable order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ with canalyzing input values $a_1, \ldots, a_n$ and canalyzed output values $b_1, \ldots, b_n$, respectively. Furthermore, any nested canalyzing function can be represented in the form (2.2)*.*

PROOF. It is clear that g depends on all variables $x_1, \ldots, x_n$. Let $g_n = x_{\sigma(n)} + a_n + b_n$ and, for $1 \le i < n$, let

$$g_i = (x_{\sigma(i)} + a_i + b_i)\Diamond_i g_{i+1}.$$

Then $g = g_1 = (x_{\sigma(1)} + a_1 + b_1)\Diamond_1 g_2$.

If $x_{\sigma(1)} = a_1$, then $(x_{\sigma(1)} + a_1 + b_1)\Diamond_1 g_2 = b_1 \Diamond_1 g_2 = b_1$, by equation (2.3). For $1 \le i \le n-1$, suppose $x_{\sigma(j)} = \overline{a_j}$ for $j < i$ and $x_{\sigma(i)} = a_i$. Now, for all $j < i$, we have $\overline{b_j}\Diamond_j g_{j+1} = g_{j+1}$ and $b_i \Diamond_i g_{i+1} = b_i$. Thus, by equation (2.3), we get

$$\overline{b_1}\Diamond_1(\overline{b_2}\Diamond_2(\ldots(b_i\Diamond_i g_{i+1})\ldots)) = b_i\Diamond_i g_{i+1} = b_i.$$

Hence g is nested canalyzing, with the a_i as canalyzing values and the b_i as canalyzed values. It is left to show that any nested canalyzing function can be represented in the form (2.2). Let f be a nested canalyzing function in the variable order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ with canalyzing input values $a_1, \ldots, a_n$ and canalyzed output values $b_1, \ldots, b_n$, respectively. By Lemma 2.5 and the above, it is clear that $f(c_1, \ldots, c_n) = g(c_1, \ldots, c_n)$ for all $(c_1, \ldots, c_n) \in \mathbb{F}_2^n$. Thus $f = g$ as functions and hence f can be represented in the form (2.2).

2.2 NCFs are Unate Cascade Functions

We next show that Boolean NCFs are equivalent to unate cascade functions. Unate cascade functions have been defined and studied [14; 16] as a special class of fanout-free functions which are used in the design and synthesis of logic circuits and switching theory [3; 6].

Definition 2.7 *A Boolean function f is a* unate cascade *function if it can be represented as*

$$f(x_1, x_2, \ldots, x_n) = x^*_{\sigma(1)}\Diamond_1(x^*_{\sigma(2)}\Diamond_2(\ldots(x^*_{\sigma(n-1)}\Diamond_{n-1}x^*_{\sigma(n)}))\ldots), \tag{2.4}$$

where σ is a permutation on $\{1, \ldots, n\}$, x^* is either x or $x+1$ and $\Diamond_i$ is either the OR ($\vee$) or AND ($\wedge$) Boolean operator.

Theorem 2.8 *A Boolean function is nested canalyzing if and only if it is a unate cascade function.*

PROOF. Let f be a unate cascade function in the form (2.4). Let a_n and b_n be such that $x^*_{\sigma(n)} = x_{\sigma(n)} + a_n + b_n$ and, for $1 \leq i < n$, let

$$b_i = \begin{cases} 1, & \text{if } \Diamond_i = \vee; \\ 0, & \text{if } \Diamond_i = \wedge, \end{cases} \tag{2.5}$$

and let $a_i \in \{0, 1\}$ such that $x^*_{\sigma(i)} = x_{\sigma(i)} + a_i + b_i$. That is,

$$a_i = x^*_{\sigma(i)} + x_{\sigma(i)} + b_i. \tag{2.6}$$

Then

$$f(x_1, \dots, x_n) = (x_{\sigma(1)}+a_1+b_1)\Diamond_1((x_{\sigma(2)}+a_2+b_2)\Diamond_2(\cdots((x_{\sigma(n-1)}+a_{n-1}+b_{n-1})\Diamond_{n-1}(x_{\sigma(n)}+a_n+b_n))\cdots),$$

which is nested canalyzing by Lemma 2.6.

Conversely, let f be a nested canalyzing function of the form (2.1). By Lemma 2.6 and equation (2.2), f can be represented in the form (2.4) where $x^*_{\sigma(i)} = x_{\sigma(i)} + a_i + b_i$ for all $1 \leq i \leq n$. Thus f is unate cascade.

Remark 2.9 *The second sentence in the proof above implies the nested canalyzing function with canalyzing input* $(a_1, \dots, a_n)$ *and canalyzed output* $(b_1, \dots, b_n)$ *is also nested canalyzing in the same variable order with canalyzing input* $(a_1, \dots, a_{n-1}, \overline{a_n})$ *and canalyzing output* $(b_1, \dots, b_{n-1}, \overline{b_n})$.

Remark 2.10 *The theorem above provides natural equivalence between the class of nested canalyzing functions and that of unate cascade functions. Namely, for a given unate cascade function in the form (2.4), using (2.6) and (2.5) we can explicitly define the canalyzing input values* a_i *and canalyzed output values* b_i*. On the other hand, any nested canalyzing function in the form (2.1) can be presented as a unate cascade function in the form (2.4) where* $x^*_{\sigma(i)} = x_{\sigma(i)} + a_i + b_i$ *and* $\Diamond_i$ *as in (2.3) for all* $1 \leq i \leq n$.

Using the theorem and remark above, it is now possible to translate results about one type of function into results about the other type. We point out one such example. Several papers have been dedicated to counting the number of certain fanout-free functions including the unate cascade functions [1; 2; 6; 17; 18]. On the other hand, Just et. al. [7], gave a formula for the number of canalyzing functions.

Bender and Butler [2] and Sasao and Kinoshita [18] independently found the number of unate cascade functions, among other fanout-free functions. As an immediate corollary of Theorem 2.8, we therefore know the number of NCFs in n variables, for a given value of n. We use the recursive formula found by Sasao and Kinoshita [18], in the following corollary.

Corollary 2.11 *The number of NCFs in n variables, denoted by $NCF(n)$, is given by*

$$NCF(n) = 2 \cdot E(n),$$

where

$$E(1) = 1, \quad E(2) = 4,$$

and, for $n \geq 3$,

$$E(n) = \sum_{r=2}^{n-1} \binom{n}{r-1} \cdot 2^{r-1} \cdot E(n-r+1) + 2^n.$$

For example, the number of Boolean NCFs (unate cascade functions) on n variables for $n \leq 8$ is given by Table 1, which is part of the tables given by Sasao and Kinoshita [18] and also Bender and Butler [2].

Table 1
The number of NCFs on $n \leq 8$ variables

n	1	2	3	4	5	6	7	8
$NCF(n)$	2	8	64	736	10,624	183,936	3,715,072	85,755,392

Some interesting facts derive from the equivalence of NCFs with unate cascade functions. Sasao and Kinoshita [18, Lemma 4.1], found that unate cascade functions are equivalent to fanout-free threshold functions. Thus NCFs are a special class of threshold functions. It is also interesting to note that among switching networks, the unate cascade functions are precisely those with the smallest average path length, as shown by Butler et. al. [3], which makes them efficient in logic circuit design.

3 Nested canalyzing functions as polynomial functions

Wanting to compute the total number of Boolean functions of a particular type, e.g. nested canalyzing or unate cascade functions, is one example of the need to study the totality of such functions. Few tools other than elementary combinatorics are available for this purpose, however. In this section, we propose an alternative approach to Boolean functions which provides a whole new set of mathematical tools and results. We will view Boolean functions $\{0,1\}^n \longrightarrow \{0,1\}$ as polynomial functions $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$, where $\mathbb{F}_2$ denotes the field with two elements. It is well-known that any function $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$ can be represented as a polynomial function [12]. If we require that every variable appear with exponent 1, then this representation is unique. For Boolean functions, this representation is straightforward to construct by observing that $\mathrm{AND}(x, y) = x \wedge y = xy$, $\mathrm{OR}(x, y) = x \vee y = x + y + xy$, and $\mathrm{NOT}(x) = \overline{x} = x + 1$. Conversely, replacing multiplication by AND and addition by the XOR function, we can translate any polynomial function into a Boolean function. In particular, this shows that any binary function on n variables can be represented as a Boolean function. While this seems like a simple change of language, it has the profound effect of placing the

study of Boolean functions into the fields of algebra and algebraic geometry, which have a rich body of results and algorithms available.

The goal of this section is to formalize this equivalence and to characterize those polynomial functions that represent nested canalyzing functions. The characterization will be expressed as a parametrization, with the set of parameters taken from an algebraic variety. Algebraic geometry has many tools to study varieties, an approach that will be pursued elsewhere.

3.1 Polynomial form of nested canalyzing functions

We derive a polynomial representation of the class of Boolean nested canalyzing functions which we then use to identify necessary and sufficient relations among their coefficients.

Any Boolean function in n variables is a map $f : \{0,1\}^n \longrightarrow \{0,1\}$. The set of all such maps, denoted by B_n, can be given the algebraic structure of a ring with the boolean operators XOR for addition and the conjunction AND for multiplication.

Consider the polynomial ring $\mathbb{F}_2[x_1,\dots,x_n]$ over the field $\mathbb{F}_2 := \{0,1\}$ with two elements. Let I be the ideal generated by the polynomials $x_i^2 - x_i$ for all $i = 1,\dots,n$. (That is, I consists of all linear combinations of these polynomials with arbitrary polynomials as coefficients.) For any Boolean function $f \in B_n$, there is a unique polynomial $g \in \mathbb{F}_2[x_1,\dots,x_n]$ such that $g(a_1,\dots,a_n) = f(a_1,\dots,a_n)$ for all $(a_1,\dots,a_n) \in \mathbb{F}_2^n$ and such that the degree of each variable appearing in g is equal to 1. Namely

$$g(x_1,\dots,x_n) = \sum_{(a_1,\dots,a_n)\in\mathbb{F}_2^n} f(a_1,\dots,a_n)\prod_{i=1}^{n}(1-(x_i-a_i)). \tag{3.1}$$

And it is straightforward to show that this equivalence extends to a ring isomorphism

$$R := \mathbb{F}_2[x_1,\dots,x_n]/I \cong B_n.$$

From now on we will not distinguish between these two rings.

Next we present and study the set of all Boolean nested canalyzing functions as a subset of the ring R of all Boolean polynomial functions.

The following theorem gives the polynomial form for canalyzing and nested canalyzing functions.

Theorem 3.1 *Let f be a function in R. Then*

(1) The function f is canalyzing *in the variable x_i, for some i, $1 \le i \le n$, with canalyzing input value a_i and canalyzed output value b_i, if and only if*

$$f(x_1,x_2,\dots,x_i,\dots,x_n) = (x_i - a_i)g(x_1,x_2,\dots,x_i,\dots,x_n) + b_i. \tag{3.2}$$

(2) *The function* f *is* nested canalyzing *in the order* $x_1, x_2, \ldots, x_n$*, with canalyzing values* a_i *and corresponding canalyzed values* b_i*,* $1 \le i \le n$*, if and only if it has the polynomial form*

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) = {} & (x_1 - a_1)[(x_2 - a_2)[\ldots[(x_{n-1} - a_{n-1})[(x_n - a_n) \\ & +(b_n - b_{n-1})] + (b_{n-1} - b_{n-2})]\ldots] + (b_2 - b_1)] + b_1 \end{aligned} \tag{3.3}$$

or, equivalently,

$$f(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n}(x_i - a_i) + \sum_{j=1}^{n-1}\left[(b_{n-j+1} - b_{n-j})\prod_{i=1}^{n-j}(x_i - a_i)\right] + b_1. \tag{3.4}$$

PROOF.

(1) It is easy to see that if $x_i = a_i$, then the output is b_i, no matter what the values of the other variables are. Conversely, if f is canalyzing with input a_i and output b_i, then $f - b_i$, as a polynomial in x_i has a_i as a root, hence is divisible by $x_i - a_i$. This proves the first claim.

(2) Let f be a nested canalyzing function as in Definition 2.3, and let

$$\begin{aligned} g(x_1, \ldots, x_n) = {} & (x_1 - a_1)[(x_2 - a_2)[\ldots[(x_{n-1} - a_{n-1})[(x_n - a_n) \\ & +(b_n - b_{n-1})] + (b_{n-1} - b_{n-2})]\ldots] + (b_2 - b_1)] + b_1. \end{aligned}$$

We will show that g is the unique polynomial presentation of f, as in equation 3.1. Since the degree of each variable in g is equal to one, we only need to show that $g(c_1, \ldots, c_n) = f(c_1, \ldots, c_n)$ for all $(c_1, \ldots, c_n) \in \mathbb{F}_2^n$.

Clearly, if $c_1 = a_1$, then $g(c_1, \ldots, c_n) = b_1$. If $c_1 \ne a_1$ and $c_2 = a_2$, then $(c_1 - a_1) = 1$ and $g(c_1, \ldots, c_n) = b_2$. If $c_1 \ne a_1$, $c_2 \ne a_2$ and $c_3 = a_3$, then $g(c_1, \ldots, c_n) = b_3$. We continue until we have $c_i \ne a_i$ for all $1 \le i < n$ and $c_n = a_n$, in which case we get $g(c_1, \ldots, c_n) = b_n$. If $c_i \ne a_i$ for all i, then $(c_i - a_i) = 1$ for all i and hence $g(c_1, \ldots, c_n) = 1 + b_n$. Thus g is the unique polynomial representation of f.

3.2 A Parametrization of NCFs

Our next goal is to derive a criterion as to when a given Boolean polynomial in n variables is a nested canalyzing function. The criterion will be given in terms of a parametrization of such polynomials corresponding to points in the affine space $\mathbb{F}_2^{2^n}$ that satisfy a certain collection of polynomial equations. Such a set is by definition an algebraic variety, in the language of algebraic geometry. This parametrization describes the entire space of nested canalyzing functions as a geometric object, whose properties can then be studied with the tools of algebraic geometry.

Recall that the ring of Boolean functions is isomorphic to the quotient ring $R = \mathbb{F}_2[x_1, \ldots, x_n]/I$, where $I = \langle x_i^2 - x_i : 1 \le i \le n \rangle$. Therefore, the terms of a Boolean polynomial consist of square-free monomials. Thus, we can uniquely index monomials by the subsets of $[n] := \{1, \ldots, n\}$ corresponding to the variables appearing in the monomial, so that we can write the elements of R as

$$R = \{ \sum_{S \subseteq [n]} c_S \prod_{i \in S} x_i \; : \; c_S \in \mathbb{F}_2 \}. \tag{3.5}$$

As a vector space over $\mathbb{F}_2$, R is isomorphic to $\mathbb{F}_2^{2^n}$ via the correspondence

$$R \ni \sum_{S \subseteq [n]} c_S \prod_{i \in S} x_i \longleftrightarrow (c_\emptyset, \ldots, c_{[n]}) \in \mathbb{F}_2^{2^n}, \tag{3.6}$$

for a given fixed total ordering of all square-free monomials. That is, a polynomial function corresponds to the vector of coefficients of the monomial summands. In this section we identify the set of nested canalyzing functions in R with a subset V^{ncf} of $\mathbb{F}_2^{2^n}$ by imposing relations on the coordinates of its elements.

Let S be any subset of $[n]$. We introduce a new term called the *completion* of S.

Definition 3.2 *Let S be a a non-empty set whose highest element is r_S. The completion of S, which we denote by $[r_S]$, is the set $[r_S] := \{1, 2, \ldots, r_S\}$. For $S = \emptyset$, let $[r_\emptyset] := \emptyset$.*

The main result of this section is the following theorem.

Theorem 3.3 *Let f be a Boolean polynomial in n variables, given by*

$$f(x_1, x_2, \ldots, x_n) = \sum_{S \subseteq [n]} c_S \prod_{i \in S} x_i. \tag{3.7}$$

The polynomial f is a nested canalyzing function in the order $x_1, x_2, \ldots, x_n$ if and only if $c_{[n]} = 1$, and for any subset $S \subseteq [n]$,

$$c_S = c_{[r_S]} \prod_{i \in [r_S] \setminus S} c_{[n] \setminus \{i\}}. \tag{3.8}$$

PROOF. First assume that the polynomial f is a Boolean nested canalyzing function in the order $x_1, x_2, \ldots, x_n$, with canalyzing input values a_i and corresponding canalyzed output values $b_i, 1 \le i \le n$. Then, by part 2 of Theorem 3.1, f has the form (3.4) which can be expanded as

$$f(x_1, \ldots, x_n) = \sum_{S \subseteq [n]} \prod_{i \in S} x_i \prod_{l \in [n] \setminus S} a_l + \sum_{j=1}^{n-1} (b_{n-j+1} - b_{n-j}) (\sum_{S \subseteq [n-j]} \prod_{i \in S} x_i \prod_{l \in [n-j] \setminus S} a_l) + b_1. \tag{3.9}$$

We now equate corresponding coefficients in equations (3.7) and (3.9). First let $S = [n]$. Then, clearly, $c_{[r_S]} = 1$. Next, consider subscripts of the form $S = [n] \setminus \{i\}, i \neq n$, that is, coefficients

of monomials of total degree $n-1$ which contain x_n. It is clear from equation (3.9) that x_n only appears in the first summand and hence, for $1 \le i \le n-1$,

$$c_{[n]\backslash\{i\}} = a_i = c_{[r_S]}c_{[n]\backslash\{i\}}. \tag{3.10}$$

It is easy to check that equation (3.8) holds for any set $S \subseteq [n]$ such that $S = [r_S]$. By equating the coefficient of $x_1 \cdots x_{r_s}$ in equations (3.9) and (3.7), we get

$$\begin{aligned} c_S = c_{[r_S]} &= \prod_{i\in[n]\backslash S} a_i + (b_n - b_{n-1}) \prod_{i\in[n-1]\backslash S} a_i + \cdots + (b_{r_S+1} - b_{r_S}) \prod_{i\in[r_S]\backslash S} a_i \\ &= \prod_{i\in[n]\backslash S} a_i + (b_n - b_{n-1}) \prod_{i\in[n-1]\backslash S} a_i + \cdots + (b_{r_S+1} - b_{r_S}), \end{aligned}$$

since $\prod_{i\in[r_S]\backslash S} a_i = \prod_{i\in\emptyset} a_i := 1$, by definition. Now let S be any nonempty index set. Then

$$\begin{aligned} c_S &= \prod_{i\in[n]\backslash S} a_i + (b_n - b_{n-1}) \prod_{i\in[n-1]\backslash S} a_i + \cdots + (b_{r_S+1} - b_{r_S}) \prod_{i\in[r_S]\backslash S} a_i \\ &= \prod_{i\in[r_S]\backslash S} a_i [\prod_{i\in[n]\backslash[r_S]} a_i + (b_n - b_{n-1}) \prod_{i\in[n-1]\backslash[r_S]} a_i + \cdots + (b_{r_S+1} - b_{r_S})] \\ &= (\prod_{i\in[r_S]\backslash S} a_i) c_{[r_S]} \\ &= c_{[r_S]} \prod_{i\in[r_S]\backslash S} c_{[n]\backslash\{i\}}. \end{aligned}$$

This completes the proof that a nested canalyzing polynomial has to satisfy equation (3.8).

Conversely, suppose that $c_{[n]} = 1$ and equation (3.8) holds for the coefficients of the polynomial f in equation (3.7). We need to show that f is nested canalyzing. Using Lemma 2.5, it is enough to show that f depends on all n variables and $f(\overline{a_1}, \ldots, \overline{a_{j-1}}, a_j, x_{j+1}, \ldots, x_n) = b_j$ for some $a_j, b_j \in \mathbb{F}_2^n$ and $1 \le j \le n$. Since $c_{[n]} = 1$, the monomial $x_1 \cdots x_n$ is a summand in f and hence f depends on all n variables. Now let $1 \le j \le n$. For any $S \subset [n]$ such that $j \notin S$ and $r_S > j$, we have

$$c_S = c_{[r_S]} \prod_{i\in[r_S]\backslash S} c_{[n]\backslash\{i\}} \quad \text{and} \quad c_{S\cup\{j\}} = c_{[r_S]} \prod_{i\in[r_S]\backslash\{S\cup\{j\}\}} c_{[n]\backslash\{i\}}.$$

By pairing c_S with $c_{S\cup\{j\}}$ and c_T with $c_{T\cup\{j\}}$ where $T \subseteq [j-1]$, we rewrite the form (3.7) into

$$f(x_1, \ldots, x_n) = \sum_{T\subseteq[j-1]} (x_j c_{T\cup\{j\}} + c_T) \prod_{i\in T} x_i + (c_{[n]\backslash\{j\}} + x_j) \sum_{\substack{S\subset[n] \\ r_S>j \\ j\notin S}} c_{S\cup\{j\}} \prod_{i\in S} x_i. \tag{3.11}$$

For $1 \le j \le n$, let $a_j = c_{[n]\backslash\{j\}}$. Then

$$f(\overline{a_1}, \ldots, \overline{a_{j-1}}, a_j, x_{j+1}, \ldots, x_n) = \sum_{T\subseteq[j-1]} (c_{[n]\backslash\{j\}} c_{T\cup\{j\}} + c_T) \prod_{i\in T} (1 + c_{[n]\backslash\{i\}}) \tag{3.12}$$

is a constant which we call b_j. Hence, by Lemma 2.5, the function f is nested canalyzing.

Remark 3.4 *Observe that the relations in equation (3.8) leave the coefficients $c_\emptyset$ and $c_{[i]}$, for all $1 \le i < n$, undetermined, as well as the coefficients c_S, where S is any of the $(n-1)$-element subsets of $[n]$ which include n. Furthermore, a Boolean NCF requires that $c_{[n]} = 1$. Since a general Boolean polynomial in n variables has 2^n coefficients, equation (3.8) yields $2^n - 2n$ equations which have to be satisfied by the coefficients of a Boolean NCF.*

Corollary 3.5 *The set of points in $\mathbb{F}_2^{2^n}$ corresponding to coefficient vectors of nested canalyzing functions in the variable order $x_1, \ldots, x_n$, denoted by $V_{\text{id}}^{\text{ncf}}$, is given by*

$$V_{\text{id}}^{\text{ncf}} = \{(c_\emptyset, \ldots, c_{[n]}) \in \mathbb{F}_2^{2^n} : c_{[n]} = 1,\ c_S = c_{[r_S]} \prod_{i \in [r_S] \setminus S} c_{[n]\setminus\{i\}},\ \textit{for}\ S \subseteq [n]\}. \tag{3.13}$$

The following corollary provides surprisingly simple expressions of the canalyzing input and canalyzed output values in terms of the coefficients of the polynomial.

Corollary 3.6 *Let f be a Boolean polynomial given by equation (3.7). If the polynomial f is a nested canalyzing function in the order $x_1, x_2, \ldots, x_n$, with input values a_j and corresponding output values $b_j, 1 \le j \le n$, then*

$$\begin{aligned}
a_j &= c_{[n]\setminus\{j\}}, && \text{for} \quad 1 \le j \le n-1 && (3.14)\\
b_1 &= c_\emptyset + c_1 c_{[n]\setminus\{1\}}, && && (3.15)\\
b_{j+1} - b_j &= c_{[j+1]} c_{[n]\setminus\{j+1\}} + c_{[j]}, && \text{for } 1 \le j < n-1 \text{ and} && (3.16)\\
b_n - a_n &= b_{n-1} + c_{[n-1]}\ . && && (3.17)
\end{aligned}$$

PROOF. Equation (3.14) follows from equation (3.10), equation (3.15) follows directly from equation (3.12) when $j = 1$. In equation (3.3), we observe that the variable x_{n-1} appears only in the first and second group of products. In particular, $c_{[n-1]} = -a_n + b_n - b_{n-1}$, and hence equation (3.17) follows.

It is left to show (3.16). From equation (3.12),

$$\begin{aligned}
b_j &= \sum_{T \subseteq [j-1]} (c_{[n]\setminus\{j\}} c_{T \cup \{j\}} + c_T) \prod_{i \in T} (1 + c_{[n]\setminus\{i\}})\\
&= \sum_{T \subseteq [j-1]} (c_{[n]\setminus\{j\}} c_{[j]} \prod_{i \in [j-1]\setminus T} c_{[n]\setminus\{i\}} + c_T) \prod_{i \in T} (1 + c_{[n]\setminus\{i\}})\\
&= c_{[n]\setminus\{j\}} c_{[j]} \sum_{T \subseteq [j-1]} c_{[n]\setminus\{j\}} c_{[j]} \prod_{i \in [j-1]\setminus T} c_{[n]\setminus\{i\}} \prod_{i \in T} (1 + c_{[n]\setminus\{i\}}) + \sum_{T \subseteq [j-1]} c_T \prod_{i \in T} (1 + c_{[n]\setminus\{i\}})\\
&= c_{[n]\setminus\{j\}} c_{[j]} + \sum_{T \subseteq [j-1]} c_T \prod_{i \in T} (1 + c_{[n]\setminus\{i\}}),
\end{aligned}$$

since

$$\sum_{T\subseteq[j-1]}\prod_{i\in[j-1]\setminus T} c_{[n]\setminus\{i\}}\prod_{i\in T}(1+c_{[n]\setminus\{i\}}) = \prod_{i\in[j-1]}(c_{[n]\setminus\{i\}}+(1+c_{[n]\setminus\{i\}})) = \prod_{i\in[j-1]} 1 = 1.$$

Now

$$\begin{aligned}
b_{j+1}-b_j &= c_{[n]\setminus\{j+1\}}c_{[j+1]} + \sum_{T\subseteq[j]} c_T\prod_{i\in T}(1+c_{[n]\setminus\{i\}}) - c_{[n]\setminus\{j\}}c_{[j]} - \sum_{T\subseteq[j-1]} c_T\prod_{i\in T}(1+c_{[n]\setminus\{i\}})\\
&= c_{[n]\setminus\{j+1\}}c_{[j+1]} - c_{[n]\setminus\{j\}}c_{[j]} + \sum_{\substack{T\subseteq[j]\\ j\in T}} c_T\prod_{i\in T}(1+c_{[n]\setminus\{i\}})\\
&= c_{[n]\setminus\{j+1\}}c_{[j+1]} - c_{[n]\setminus\{j\}}c_{[j]} + \sum_{T\subseteq[j-1]}(1+c_{[n]\setminus\{j\}})c_{[j]}\prod_{i\in[j-1]\setminus T} c_{[n]\setminus\{i\}}\prod_{i\in T}(1+c_{[n]\setminus\{i\}})\\
&= c_{[n]\setminus\{j+1\}}c_{[j+1]} - c_{[n]\setminus\{j\}}c_{[j]} + (1+c_{[n]\setminus\{j\}})c_{[j]}\sum_{T\subseteq[j-1]}\prod_{i\in[j-1]\setminus T} c_{[n]\setminus\{i\}}\prod_{i\in T}(1+c_{[n]\setminus\{i\}})\\
&= c_{[n]\setminus\{j+1\}}c_{[j+1]} - c_{[n]\setminus\{j\}}c_{[j]} + (1+c_{[n]\setminus\{j\}})c_{[j]}\\
&= c_{[n]\setminus\{j+1\}}c_{[j+1]} + c_{[j]}
\end{aligned}$$

Remark 3.7 *Equations (3.14)–(3.16) imply that the input values a_i and output values b_i, $1\le i\le n-1$, are determined uniquely by the coefficients of the polynomial f. Also, equation (3.17) implies that there are two sets of values for a_n and b_n which will yield the same nested canalyzing function f. Using these facts, we discover Remark 2.9.*

Example 3.8 *In Table 2, we give some examples of relationships between coefficients of NCFs in n variables, nested in the order $x_1, x_2, \dots, x_n$ for some small values of n.*

Table 2
Some examples of relationships between coefficients of NCFs

$n=3$	$n=4$	$n=5$
$c_3 = c_{13}c_{23}c_{123}$	$c_4 = c_{234}c_{134}c_{124}c_{1234}$	$c_5 = c_{2345}c_{1345}c_{1245}c_{1235}c_{12345}$
$c_2 = c_{23}c_{12}$	$c_{13} = c_{134}c_{123}$	$c_{124} = c_{1245}c_{1234}$
	$c_{24} = c_{234}c_{124}c_{1234}$	$c_{23} = c_{2345}c_{123}$
	$c_2 = c_{234}c_{12}$	$c_2 = c_{2345}c_{12}$

We now extend Theorem 3.3 to the general case when the variables are nested in any given order. For this, we will need to extend the definition of *completion* of a set S with respect to any permutation of its elements.

Definition 3.9 *Let σ be a permutation on the elements of the set $[n]$. We define a new order relation $<_\sigma$ on the elements of $[n]$ as follows: $i <_\sigma j$ if and only if $\sigma^{-1}(i) < \sigma^{-1}(j)$. Let S be a nonempty subset of $[n]$, say $S = \{i_1, \dots, i_t\}$. Let $r_S^\sigma := \max\{\sigma^{-1}(i_1), \dots, \sigma^{-1}(i_t)\}$.*

The completion of S with respect to the permutation σ, *denoted by* $[r_S^\sigma]_\sigma$, *is the set* $[r_S^\sigma]_\sigma := \{\sigma(1), \ldots, \sigma(r_S^\sigma)\}$.

The following corollary is a generalization of Theorem 3.3. It gives necessary and sufficient relations among the coefficients of a NCF whose variables are nested in the order specified by a permutation σ on $[n]$.

Corollary 3.10 *Let $f \in R$ and let σ be a permutation of the set $[n]$. The polynomial f is a nested canalyzing function in the order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$, with input values $a_{\sigma(i)}$ and corresponding output values $b_{\sigma(i)}, 1 \le i \le n$, if and only if $c_{[n]} = 1$ and, for any subset $S \subseteq [n]$,*

$$c_S = c_{[r_S^\sigma]_\sigma} \prod_{w \in [r_S^\sigma]_\sigma \setminus S} c_{[n]\setminus\{w\}}. \tag{3.18}$$

PROOF. We follow the same argument as in the proof of Theorem 3.3, where we impose the order relation $<_\sigma$ on the elements of $[n]$ and we replace all occurrences of the subscript i by $\sigma(i)$ and $[r_S]$ by $[r_S^\sigma]_\sigma$.

Corollary 3.11 *Let σ be a permutation on $[n]$. The set of points in $\mathbb{F}_2^{2^n}$ corresponding to nested canalyzing functions in the variable order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$, denoted by V_σ^{ncf}, is defined by*

$$V_\sigma^{\text{ncf}} = \{(c_\emptyset, \ldots, c_{[n]}) \in \mathbb{F}_2^{2^n} : c_{[n]} = 1,\ c_S = c_{[r_S^\sigma]_\sigma} \prod_{w \in [r_S^\sigma]_\sigma \setminus S} c_{[n]\setminus\{w\}}, \quad \textit{for } S \subseteq [n]\}. \tag{3.19}$$

The following corollary is an extension of Corollary 3.6 and gives the input and output values of a Boolean NCF whose variables are nested in the order specified by some permutation σ.

Corollary 3.12 *Let $f \in R$ and let σ be a permutation of the elements of the set $[n]$. If f is a nested canalyzing function in the order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$, with input values a_j and corresponding output values $b_j, 1 \le j \le n$, then*

$$\begin{aligned}
a_j &= c_{[n]\setminus\{\sigma(j)\}}, && \text{for } \ 1 \le j \le n-1 && (3.20)\\
b_1 &= c_\emptyset + c_{\sigma(1)} c_{[n]\setminus\{\sigma(1)\}}, && && (3.21)\\
b_{j+1} - b_j &= c_{[j+1]_\sigma} c_{[n]\setminus\{\sigma(j+1)\}} + c_{[j]_\sigma}, && \text{for } \ 1 \le j < n-1 \quad \text{and} && (3.22)\\
b_n - a_n &= b_{n-1} + c_{[n-1]_\sigma}. && && (3.23)
\end{aligned}$$

PROOF. This follows from Corollary 3.6, where we replace all occurrences of subscript j by $\sigma(j)$ and $[r]$ by $[r]_\sigma$.

Recall that the set V^{ncf} of nested canalyzing functions is the union of the sets V_σ^{ncf} of canalyzing functions with respect to a specified variable order. By Corollaries 3.5, 3.11, and the correspondence (3.6), we have

$$V^{\mathrm{ncf}} = \bigcup_{\sigma} V_{\sigma}^{\mathrm{ncf}}.$$

Corollary 3.10 is the starting point for a geometric analysis of the set of all nested canalyzing functions. It provides a set of equations that have to be satisfied by the coefficient vectors of the polynomial representations of the functions. These coefficient vectors therefore form an algebraic variety in the space $\mathbb{F}_2^{2^n}$, which turns out to have very nice properties. In particular, it is a so-called toric variety.

4 Discussion

Our main contribution in this paper is to connect three different fields of inquiry into Boolean functions, which were heretofore apparently unconnected. The equivalence of nested canalyzing functions and unate cascade functions relates the electrical engineering point of view of logic circuits with the dynamic biological network view, providing a dictionary for results. The equivalence of both to a class of polynomial functions brings rich additional mathematical structure to the study of both. In particular, the language and concepts of algebraic geometry and the rich tool set of computational algebra and algebraic geometry provides a foundation that imposes a mathematical structure on the entire class of these functions, which suggests an entirely new way of studying them. As an algebraic variety, the class of nested canalyzing functions has a very special structure, namely that of a toric variety [4]. Toric varieties lie at the interface of geometry, algebra, and combinatorics and have a rich structure [21]. In another paper, we will explore the properties of the toric varieties in the previous section in more detail.

In particular, our motivation for this study originally was the desire to give a characterization of nested canalyzing functions as polynomials, which could be used as part of the model selection algorithm in [11]. That is, we are interested in giving an efficient criterion which allows our symbolic computation algorithm to preferentially pick nested canalyzing functions rather than general polynomials. The characterization of this class as a toric variety is the first important step in this direction.

It deserves mention that the connection to unate cascade functions was discovered in a roundabout way. We first established the parametrization of nested canalyzing functions by special polynomials. The structure of these polynomials makes it easy to count how many there are for a given number of variables. After carrying out this counting procedure for the first few numbers resulted in a sequence of integers which we submitted to N. Sloane's integer sequence database (`http://www.research.att.com/~njas/sequences/`). One of the matching sequences was that for the number of unate cascade functions.

5 Acknowledgements

Jarrah and Laubenbacher were supported partially by NSF Grant DMS-0511441. Laubenbacher was also supported partially by NIH Grant RO1 GM068947-01, a joint computational biology initiative between NIH and NSF. Raposa was supported by a Fulbright research grant while visiting the Virginia Bioinformatics Institute (VBI) where this research was conducted. She thanks VBI for the hospitality during her stay.

References

[1] Bender, E. A.: 1980, 'The Number of Fanout-Free Functions with Various Gates'. *J. Assoc. Comput. Machinery* **27**(1), 181–190.

[2] Bender, E. A. and J. T. Butler: 1978, 'Asymptotic Aproximations for the Number of Fanout-Free Functions.'. *IEEE Trans. Computers* **27**(12), 1180–1183.

[3] Butler, J. T., T. Sasao, and M. Matsuura: 2005, 'Average Path Length of Binary Decision Diagrams'. *IEEE Transactions on Computers* **54**(9), 1041–1053.

[4] Fulton, W.: 1993, *Introduction to toric varieties*, Vol. 131 of *Annals of Mathematics Studies*. Princeton, NJ: Princeton University Press.

[5] Gabbouj, M., P.-T. Yu, and E. J. Coyle: 1992, 'Convergence behavior and root signal sets of stack filters'. *Circuits Syst. Signal Process.* **11**(1), 171–193.

[6] Hayes, J. P.: 1976, 'Enumeration of Fanout-Free Boolean Functions'. *J. ACM* **23**(4), 700–709.

[7] Just, W., I. Shmulevich, and J. Konvalina: 2004, 'The number and probability of canalizing functions'. *Physica D Nonlinear Phenomena* **197**, 211–221.

[8] Kauffman, S., C. Peterson, B. Samuelsson, and C. Troein: 2003, 'Random Boolean network models and the yeast transcriptional network'. *PNAS* **100**(25), 14796–14799.

[9] Kauffman, S., C. Peterson, B. Samuelsson, and C. Troein: 2004, 'Genetic networks with canalyzing Boolean rules are always stable'. *PNAS* **101**(49), 17102–17107.

[10] Kauffman, S. A.: 1993, *The Origins of Order: Self–Organization and Selection in Evolution*. New York; Oxford: Oxford University Press.

[11] Laubenbacher, R. and B. Stigler: 2004, 'A computational algebra approach to the reverse-engineering of gene regulatory networks'. *Journal of Theoretical Biology* **229**, 523–537.

[12] Lidl, R. and H. Niederreiter: 1997, *Finite fields*, Vol. 20 of *Encyclopedia of Mathematics and its Applications*. Cambridge: Cambridge University Press, second edition.

[13] Lynch, J. F.: 1995, 'On the threshold of chaos in random Boolean cellular automata'. *Random Struct. Algorithms* **6**(2-3), 239–260.

[14] Maitra, K.: 1962, 'Cascaded switching networks of two-input flexible cells'. *IRE Trans. Electron Comput.* **EC-11**, 136–143.

[15] Moreira, A. A. and L. A. Amaral: 2005, 'Canalizing Kauffman Networks: Nonergodicity and Its Effect on Their Critical Behavior'. *Physical Review Letters* **94**(21), 218702.

[16] Mukhopadhyay, A.: 1969, 'Unate Cellular Logic'. *IEEE Trans. on Comput* **18**(2), 114–121.

[17] Pogosyan, G.: 1999, 'The Number of Cascade Functions'. *Electronic Ed.* **00**, 131.

[18] Sasao, T. and K. Kinoshita: 1979, 'On the Number of Fanout-Free Functions and Unate Cascade Functions.'. *IEEE Trans. Computers* **28**(1), 66–72.

[19] Stauffer, D.: 1987, 'On forcing functions in Kauffman's random Boolean networks'. *Journal of Statistical Physics* **46**(3-4), 789–794.

[20] Stern, M. D.: 1999, 'Emergence of homeostasis and "noise imprinting" in an evolution model'. *PNAS* **96**(19), 10746–10751.

[21] Sturmfels, B.: 1996, *Gröbner bases and convex polytopes*, Vol. 8 of *University Lecture Series*. Providence, RI: American Mathematical Society.

[22] Waddington, C. H.: 1942, 'Canalisation of development and the inheritance of acquired characters'. *Nature* **150**, 563–564.

[23] Wendt, P., E. Coyle, and N. Gallagher: 1986, 'Stack Filters'. *IEEE Trans. Acoust., Speech, Signal Processing* **34**, 898–911.

[24] Yu, P.-T. and E. Coyle: 1990, 'Convergence behavior and N-roots of stack filters'. *IEEE Trans. Acoust., Speech, Signal Processing* **38**(9), 1529–1544.